Aggregated Residual Transformations for Deep Neural Networks

Saining Xie et.al

wyuzyf,April 2019

1 Introduction

提到在视觉任务中从设计特征变成了设计 network。Inception Module 是 split-transform-merge 策略，虽然精度可以，但不容易为新的数据和任务重新改进架构，因为有很多的超参和其它因素需要注意。

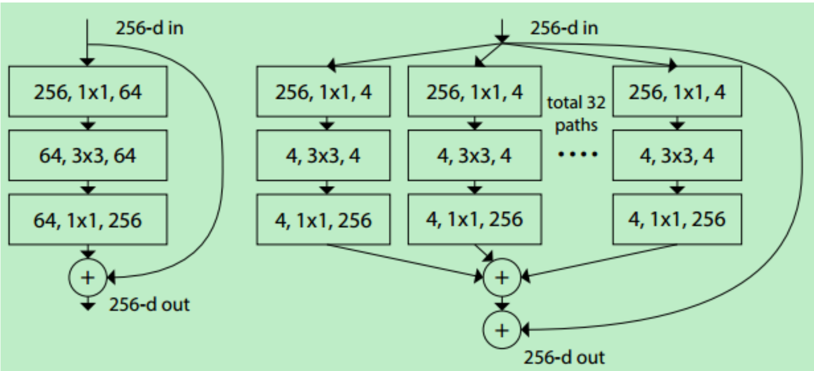

Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

图 1: 结构

Experiments demonstrate that increasing **cardinality** is a more effective way of gaining accuracy than going deeper or wider。作者将这种结构命名为 ***ResNeXt***,

2 Related Work

Multi-branch convolutional networks：Inception module 是多分支结构，Resnet 是 two-branch 结构。
Grouped convolutions：没有比 Alexnet 更早的。
Compressing convolutional networks：压缩卷积网络。
Ensembling

3 Method

- **Template**

模板结构如下：

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**$\times10^6$	**25.0**$\times10^6$
FLOPs		**4.1**$\times10^9$	**4.2**$\times10^9$

Table 1. (**Left**) ResNet-50. (**Right**) ResNeXt-50 with a 32×4d template (using the reformulation in Fig. 3(c)). Inside the brackets are the shape of a residual block, and outside the brackets is the number of stacked blocks on a stage. "C=32" suggests grouped convolutions [24] with 32 groups. *The numbers of parameters and FLOPs are similar between these two models.*

图 2: ResNeXt

- **Revisiting Simple Neurons**

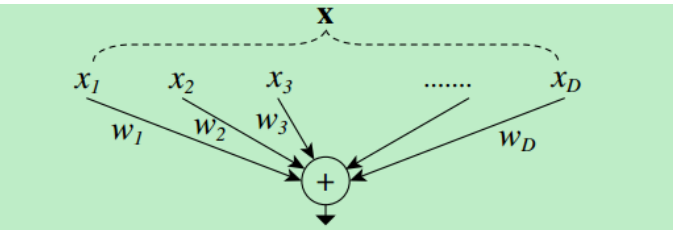

Figure 2. A simple neuron that performs inner product.

图 3: 简单的内积形式

The above operation can be recast as a combination of *splitting, transforming, and aggregating*. (i) *Splitting*: the vector $\mathbf{x}$ is sliced as a low-dimensional embedding, and in the above, it is a single-dimension subspace x_i. (ii) *Transforming*: the low-dimensional representation is transformed, and in the above, it is simply scaled: $w_i x_i$. (iii) *Aggregating*: the transformations in all embeddings are aggregated by $\sum_{i=1}^{D}$.

图 4: split-transform-merge 结构

- **Aggregated Transformations**

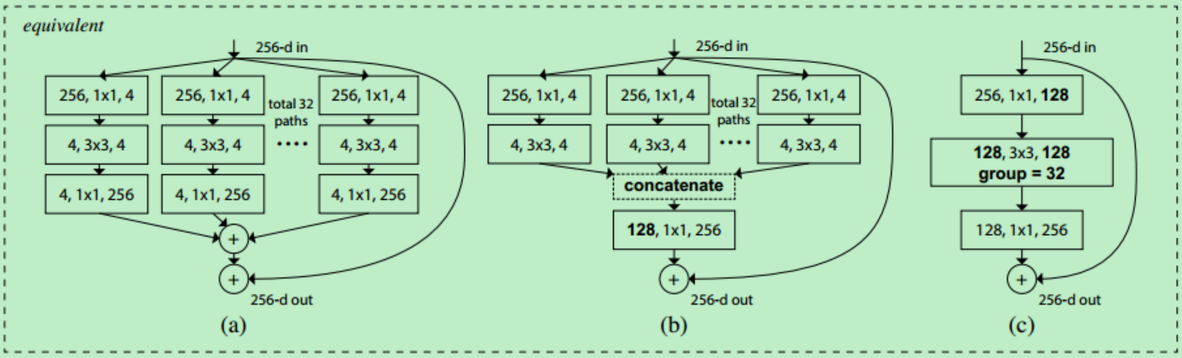

Figure 3. Equivalent building blocks of ResNeXt. **(a)**: Aggregated residual transformations, the same as Fig. 1 right. **(b)**: A block equivalent to (a), implemented as early concatenation. **(c)**: A block equivalent to (a,b), implemented as grouped convolutions [24]. Notations in **bold** text highlight the reformulation changes. A layer is denoted as (# input channels, filter size, # output channels).

图 5: 等效的 ResNeXt

b 和 c 的区别在于：b 是先 concat（即每组 4 个 channel，32*4=128，总共 128channel，这样可以减少参数，降低计算量），然后再卷积；c 是每组 128 个 channel，将其直接相加（即 32 组相同位置的 channel 直接叠加），最终也是 128channel，然后再做卷积。

总的来说，*ResNeXt* 借鉴了 *VGG/Resnet* 的 *repat layer*，以及 *group convolution* 和 *shortcut* 的思想，使得在增加深度和宽度的情况下，能够有较少的参数量，却有较好的精度表现。